AN ANALYSIS OF THE SETS OF RANK DISTRIBUTION IN AN HIERARCHICAL ORGANIZATION

by

Robert Leo Armacost

United States
Naval Postgraduate School

THESIS

AN ANALYSIS OF THE SETS
OF RANK DISTRIBUTION
IN AN HIERARCHICAL ORGANIZATION

by

Robert Leo Armacost

September 1970

An Analysis of the Sets of Rank Distribution In an Hierarchical Organization

by

Robert Leo Armacost
Lieutenant, United States Coast Guard
B.S., United States Coast Guard Academy, 1964

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
September 1970

ABSTRACT

A Markov type model for studying rank distribution in an hierarchical organization is examined. Various sets of rank distributions are defined and their properties discussed. A computer aided test is developed for testing a given rank distribution to determine if it is an element of a particular set. For a distribution not in the Steady State set, a test is developed to determine if the distribution can be returned to in m transitions or steps.

TABLE OF CONTENTS

Blank Pg.

LIST OF FIGURES

Blank

P. 6

LIST OF TABLES

Blank

P. 8

I. INTRODUCTION

Several authors have studied models for analyzing rank distributions in an hierarchical organization. Bartholomew [1967] uses Markov models to a great extent in his analysis of various social systems. Thonstad [1968] developed a Markov type model for use with the Norwegian educational system. More recently, Branchflower [1970] utilized a Markov type model to examine the promotion and appointment policies of a university and the resulting rank distributions of faculty. Branchflower assumed the promotion policies were fixed and used his model to predict the faculty distribution in the next time period using the current faculty distributions and actual appointments to each rank. In his analysis, Branchflower defines three sets of rank distributions: (1) an Attainable set, (2) the Maintainable set, and (3) the Containment set. An Attainable set is a set of rank distributions which can be reached from some previously attainable set. The Maintainable set is better known as the Steady State set of rank distributions. The Containment set is the set of rank distributions which can be returned to infinitely often by means of a sequence of appointment policies.

The analysis in Branchflower [1970] leads us to pose several questions about the rank distributions given some promotion policy:

1. Can the present rank distribution be maintained for all time by means of appointment policies alone?

2. Can another given rank distribution be attained from the present distribution in a finite time period, or can the desired distribution never be reached?

3. What are the rank distributions which can be returned to at some time?

In this paper, our purpose is first, to define explicitly the sets of rank distributions of interest associated with the model, and to demonstrate various properties of these sets. Second, we shall use these definitions and properties to develop test procedures to determine if a given distribution is an element of a particular set. This analysis will provide answers to questions 1 and 2, and will permit us to give crude bounds as an approximation to answer question 3.

II. THE BASIC MODEL AND NOTATION

Consider a discrete time Markov model with n+1 states where state j is the rank of a member of an hierarchical organization. The set of states can be partitioned into two subsets: the first n states are transient states and are called active ranks; the second subset is the $n+1^{st}$ state which is an absorbing state. Let p_{ij} be the fraction in state j at time t+1 that were in state i at time t. Then

$$\sum_{j=1}^{n+1} p_{ij} = 1, \ i=1,2,\ldots,n+1 \ .$$

We represent these fractions in a transition matrix

$$Q = \begin{bmatrix} P & w \\ 0 & 1 \end{bmatrix}$$

where P is an n x n matrix, $[p_{ij}]$, $i,j=1,2,\ldots,n$, and w is an n dimensional column vector called the loss vector with $w_i = p_{i,n+1}$. P is called the Promotion matrix and describes the transition process of the active ranks in the organization. It has the property that $\sum_{j=1}^{n} p_{ij} \leq 1$, $i=1,2,\ldots,n$ with strict inequality holding for at least one row.

We assume that the organization is in an equilibrium condition, and thus keep the total number in the organization constant. We normalize by assuming this number is one.

Let x(t) be an n dimensional row vector, $x(t) = (x_1(t), x_2(t),\ldots,x_n(t))$ where $x_i(t)$ is the fraction in the ith rank

at time t. Note that $\sum_{i=1}^{n} x_i(t) = 1$, $x_i(t) \geq 0$. Let f(t) be an n dimensional row vector, $f(t) = (f_1(t),\ldots,f_n(t))$ where $f_i(t)$ is the fractional number of new appointments to rank i at time t, $f_i(t) \geq 0$.

The rank distribution at time t+1 is dependent on the rank distribution at time t, the Promotion matrix, and the appointment policy at time t+1. In vector notation,

$$x(t+1) = x(t) \cdot P + f(t+1). \tag{1}$$

The requirement that the total number in the organization be constant requires that the total fraction appointed to the active ranks be equal to the total fraction lost to the absorbing state:

$$\sum_{i=1}^{n} f_i(t+1) = x(t) \cdot w. \tag{2}$$

We use the following notation throughout the remainder of the paper: e is an n dimensional vector with all components equal to 1; e_j is an n dimensional unit vector with the j^{th} component equal to 1. It is apparent by the context whether they are row or column vectors.

III. SET DEFINITIONS

The set of all rank distributions, F, is defined as

$$F \equiv \{x;\ x \cdot e = 1,\ x \geq 0\} \quad .$$

Note that the time notation is omitted when it does not affect the definition.

An Attainable set is a set of rank distributions which can be reached in one step from a previous set of rank distributions. Let m be the number of steps (time periods) from the set of all rank distributions, F, then

$$A_m \equiv \{x;\ x \geq z \cdot P,\ x \in F,\ z \in A_{m-1}\}\ ,\ m = 1,2,\ldots \quad \text{and}$$

$A_0 = F$.

Consider the sequence of Attainable sets, $\{A_m\}$. If the sequence converges to a set C we call the limiting set the Containment set. Mathematically, we have

$$\begin{aligned} C &\equiv \lim_{m \to \infty} A_m \\ &= \lim_{m \to \infty} \{x;\ x \geq z \cdot P,\ x \in F,\ z \in A_{m-1}\} \\ &= \{x;\ x \geq z \cdot P,\ x \in F,\ z \in C\}, \quad \text{if the limit exists.} \end{aligned}$$

This definition is somewhat unusual in that the Containment set is defined in terms of itself.

The set of all distributions which can be maintained in successive time periods is called the Steady State set, M. Let $f(t) = f$ for every t, then since $x(t+1) = x(t)$, from

(1) we have $x(t) = x(t)\cdot P + f$

$$x(t)\cdot(I-P) = f \geq 0$$

so we define

$$M \equiv \{x;\ x\cdot(I-P) \geq 0,\ x \in F\} \quad .$$

If a rank distribution can be reached from the Steady State set, M, then we call a set of such distributions a Reachable set. All distributions which are reachable from M in m steps are in the set B_m, where

$B_m \equiv \{x;\ x \geq z\cdot P,\ x \in F,\ z \in B_{m-1}\}$, $m = 1,2,\ldots$ and $B_0 = M$.

Now consider the sequence of Reachable sets, $\{B_m\}$. If this sequence converges to a set B, we call the limiting set the Reached set. Mathematically, we have

$$\begin{aligned} B &\equiv \lim_{m \to \infty} B_m \\ &= \lim_{m \to \infty} \{x;\ x \geq z\cdot P,\ x \in F,\ z \in B_{m-1}\} \\ &= \{x;\ x \geq z\cdot P,\ x \in F,\ z \in B\} \text{ , if the limit exists.} \end{aligned}$$

Thus by their definitions, we see that the Containment set and the Reached set must be the same set when they both exist. In section IV we prove they always exist.

IV. SET PROPERTIES

In this section, we list several properties of the sets defined in section III without proofs. We then prove several lemmas and two Theorems which establish the existence of the Containment set and Reached set.

PROPERTY 1: The set of all rank distributions, F, is convex and closed.

PROPERTY 2: Any Attainable set, A_m, is convex and closed.

PROPERTY 3: The Steady State set, M, is convex and closed.

PROPERTY 4: Any Reachable set, B_m, is convex and closed.

LEMMA 1: An Attainable set, A_m, is a subset of every previously attainable set.

Proof: $F = A_0 = \{x;\ x \cdot e = 1,\ x \geq 0\}$

$$A_1 = \{x;\ x \geq z \cdot P,\ x \in F,\ z \in A_0\}$$

Pick $x \in A_1 \Rightarrow x \in A_0$ therefore $A_1 \subseteq A_0 = F$

Assume $A_{m-1} \subseteq A_{m-2} \subseteq \ldots \subseteq A_1 \subseteq A_0$.

Define $A_m(z) \equiv \{x;\ x \geq z \cdot P,\ x \in F\}$, $z \in A_{m-1}$ so

$A_m = \bigcup_{z \in A_{m-1}} A_m(z)$. Pick $z^* \in A_{m-1}$, then since $A_{m-1} \subseteq A_{m-2}$,

$A_m(z^*) = \{x;\ x \geq z^* \cdot P,\ x \in F,\ z^* \in A_{m-2}\}$, and

$$A_m(z^*) \subseteq A_{m-1}.$$

So $A_m = \bigcup_{z \in A_{m-1}} A_m(z) \subseteq A_{m-1}$.

LEMMA 2: The Steady State set, M, is a subset of any Attainable set, A_m.

Proof: $M = \{x;\ x \geq x \cdot P,\ x \in F\}$. $M \subseteq A_0 = F$ by definition.

Assume $M \subseteq A_{m-1}$. Pick $z^* \in M \Rightarrow z^* \in A_{m-1}$ and $z^* \geq z^* \cdot P$.

Therefore, $z^* \in A_m$ by definition of A_m, and $M \subseteq A_m$.

LEMMA 3: Any Reachable set, B_m, contains every previously reachable set.

Proof: $M = B_0 = \{x;\ x \geq z \cdot P,\ x \in F\}$. Trivially, $B_0 \subseteq B_1$.

Assume that $B_{m-2} \subseteq B_{m-1}$ where $B_{m-1} = \{x;\ x \geq z \cdot P,$

$x \in F,\ z \in B_{m-2}\}$.

Define the following sets: $B_{m-1}(z) \equiv \{x;\ x \geq z \cdot P,$ $x \in F\}$, $z \in B_{m-2}$, and $B_m(z) \equiv \{x;\ x \geq z \cdot P,\ x \in F\}$, $z \in B_{m-1}$. Pick $z^* \in B_{m-2}$. Since $B_{m-2} \subseteq B_{m-1}$, $z^* \in B_{m-1}$, so that $B_{m-1}(z^*) = B_m(z^*)$. Since z^* is arbitrary, and since $B_{m-2} \subseteq B_{m-1}$, $B_{m-1} = \bigcup_{z \in B_{m-1}} B_{m-1}(z) \subseteq \bigcup_{z \in B_{m-1}} B_m(z)$ $= B_m$.

LEMMA 4: If $w_i > 0$ for at least one i and $n > 1$, then the Steady State set, M, contains an infinite number of points.

Proof:

(i) Non-empty. Assume that M is empty. Then there does not exist an x such that $x \cdot (I-P) \geq 0$. Because

$\sum_{j=1}^{n} p_{ij} \leq 1$, $i=1,2,\ldots,n$ with strict inequality holding

for at least one i, $(I-P)^{-1}$ exists with all elements non-negative. Let $N = (I-P)^{-1}$. If M is empty, then there does not exist an x such that $x = f \cdot N \geq 0$. But

$x_j = \sum_{i=1}^{n} f_i \cdot n_{ij}$ and since $f_i \geq 0$ and $n_{ij} \geq 0$ for every

i,j, $x_j \geq 0$ and therefore M is non-empty.

(ii) Infinite Interior. Assume that M contains only one point, then $x(t+1) = x(t)$ for every t and any appointment vector. We rewrite equation (1) as $x(t) = x(t)\cdot P + f(t+1) = f(t+1)\cdot(I-P)^{-1}$. We also require that $f(t+1)\cdot e = x(t)\cdot w$. We can construct the appointment vector in the following manner: let $f(t+1) = \alpha_1 e_1 + \alpha_2 e_2 + \ldots + \alpha_n e_n$ where $\Sigma\alpha_i = x(t)\cdot w$. Then $f(t+1) = (\alpha_1,\alpha_2,\ldots,\alpha_n)$ and

$$x(t) = (\alpha_1,\alpha_2,\ldots,\alpha_n)\cdot(I-P)^{-1}. \tag{3}$$

Consider the appointment vector $f(t+1)$ when $\alpha_1 = x(t)\cdot w$ and $\alpha_2 = \alpha_3 = \ldots = \alpha_n = 0$, then $x(t)$ equals $x(t)\cdot w$ multiplying the first row of $(I-P)^{-1}$. Since any f which solves (3) generates a point in M, consider the appointment vector $f(t+1)$ when $\alpha_2 = x(t)\cdot w$ and $\alpha_1 = \alpha_3 = \ldots = \alpha_n = 0$, then $x(t)$ equals $x(t)\cdot w$ multiplying the second row of $(I-P)^{-1}$. Since $x(t)$ is one point, and $x(t)\cdot w$ is a scalar, the first two rows of $(I-P)^{-1}$ are identical. This implies that $r((I-P)^{-1}) < n$ and that $((I-P)^{-1})^{-1}$ does not exist which is a contradiction. Thus since M is non-empty, M contains more than one point. Since M is convex and closed, it contains an infinite number of points.

THEOREM 1: The Containment Set, C, and the Reached Set, B, exist, are identical, and non-empty.

Proof: Consider the sequence of sets $\{A_m\}$. By Lemmas 1 and 2, $M \subseteq \ldots \subseteq A_m \subseteq A_{m-1} \subseteq \ldots \subseteq A_1 \subseteq F$. By Lemma 4 $M \neq \phi \Rightarrow A_m \neq \phi$, $n = 1,2,\ldots$. Thus we have a monotone decreasing sequence of sets which is bounded below, therefore the limiting set does exist and is non-empty. Now consider the sequence of sets $\{B_m\}$ where $B_0 = M$. By Lemma 3, $M \subseteq B_0 \subseteq B_1 \subseteq \ldots \subseteq B_{m-1} \subseteq B_m \subseteq \ldots \subseteq F$. Thus we have a monotone increasing sequence of sets bounded above by F. Therefore a limiting set exists. As noted in section III the Containment set, C, and the Reached set, B, are the same set.

THEOREM 2: The Steady State set, M, any Attainable set, A_m, the Containment set, C, any Reachable set, B_m, and the set of all rank distributions, F, are identical if and only if the Promotion matrix, P, is diagonal.

Proof: Since $M \subseteq B_m \subseteq C \subseteq A_m \subseteq F$, we need only show that $M = F$.

(i) Assume that P is diagonal. We can write the distribution vector as $x(t) = \alpha_1 e_1 + \alpha_2 e_2 + \ldots + \alpha_n e_n$, where $\alpha_i \geq 0$. If $\sum_i \alpha_i = 1$ then $x(t) \in F$. Rewrite equation (1) $x(t + 1) = x(t)\cdot P + f(t + 1)$. We require $f(t + 1)\cdot e = x(t)\cdot(I-P)\cdot e$. Since P is diagonal, $x(t)\cdot(I-P)$ is a suitable choice for $f(t + 1)$, $(f_i(t + 1) \geq 0)$. So we can now write $x(t + 1) = x(t)\cdot P + x(t)\cdot(I-P) = x(t)$. Now $x(t + 1) = x(t)$ and $x(t) \in F \Rightarrow x(t) \in M$, $\therefore F \subseteq M$. And since $M \subseteq F$, $M = F$.

(ii) Assume $M = F$. $x(t + 1) = x(t)$ for all t.

Assume that P is not diagonal. $x(t) = x(t) \cdot P + f(t + 1)$.

since $x(t) \in F$, we can write $x(t) = \alpha_1 e_1 + \alpha_2 e_2 + \ldots + \alpha_n e_n$, $\alpha_i \geq 0$, $\sum_i \alpha_i = 1$. In particular, let $\alpha_1 = 1$, $\alpha_i = 0$, $i \neq 1$. So $x(t) = \alpha_1 e_1$ and $(1,0,\ldots,0) =$

$$(1,0,\ldots,0) \cdot \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1n} \\ \vdots & & & \vdots \\ p_{n1} & p_{n2} & \cdots & p_{nn} \end{bmatrix} + f(t + 1)$$

$= (p_{11}, p_{12}, \ldots, p_{1n}) + (f_1, f_2, \ldots, f_n)$. This requires that the following equations be satisfied:

$$\begin{array}{ccccc} f_1 & + & p_{11} & = & 1 \\ f_2 & + & p_{12} & = & 0 \\ \vdots & & \vdots & & \vdots \\ f_n & & p_{1n} & = & 0 \end{array} .$$

Since $f_j \geq 0$, $p_{ij} \geq 0$, we have $f_j = p_{1j} = 0$, $j \neq 1$. We repeat this procedure for $\alpha_j = 1$, $\alpha_k = 0$, $k \neq j$, $j = 2,3,\ldots,n$. Thus $p_{ij} = 0$, $i \neq j$ and P is diagonal.

V. EXTREME POINTS OF THE VARIOUS SETS

The characterizations of the various sets in the model in the above section do not describe them in a useful manner. If we knew the extreme points of all of the sets, then we would have almost all of the information about the sets, since any interior point of a set is simply the convex combination of its extreme points.

For two sets in particular, the extreme points are easily obtained. The set of all rank distributions, F, is the fundamental simplex in E_n with extreme points, e_i, $i = 1,2,...n$. The subset of F which is most easily described is the Steady State set, M. Each vector in the set satisfies the inequality $x \geq x \cdot P$. This can be written in equation form as $x \cdot (I-P) = f$ where f is the vector of appointments to the different ranks. We define $N = (I-P)^{-1}$ so we write $x = f \cdot N$. Since we require $x \cdot e = 1$ and since f can be written $f = \sum_i \alpha_i (x \cdot (I-P) \cdot e)$, $\alpha_i \geq 0$, $\sum_i \alpha_i = 1$, we write the equation in an equivalent form as $x = f \cdot B$ where $f = \sum_i \alpha_i e_i$ and $B_{ij} = \frac{N_{ij}}{\sum_j N_{ij}}$. That is, the extreme points of the Steady State set are given by the rows of $(I-P)^{-1}$ after the rows are normalized.

The problem of determining the extreme points of the other sets in the model is much more difficult. If we are working in three dimensions, it is not too difficult to calculate all of the extreme points of one of the sets, A_m or B_m, for a small m.

Each set in the model has the property that each point in the set generates a subsimplex in the fundamental simplex. The extreme points of the generated set are found from the convex hull of the subsimplices generated by the extreme points of the original set. In three dimensions we use a geometrical method to determine the extreme points of the convex hull of the generated subsimplices. If we know the extreme points of A_m for some m, then we project A_m onto a plane generated by two unit vectors. Then we calculate the extreme points of the subsimplex generated by each extreme point of A_m and project these onto the plane. Then we simply test linear combinations of the generated extreme points to determine that there are no points which lie beyond the linear combination in each of the unit directions. The points which satisfy this criterion are the extreme points of the set A_{m+1}. A similar procedure is used for B_m.

When P is upper triangular, some of the extreme points of A_m are also extreme points of A_{m+1} (e.g. e_n is an extreme point of A_m for every m.) These remain extreme points for all succeeding sets generated from A_{m+1}. Similarly, B_m has extreme points common to all B_m which are extreme points of A_m as well, and therefore are extreme points of the Containment set, C. Figure 1 depicts several attainable sets drawn on barycentric coordinates for a three dimensional rank distribution where

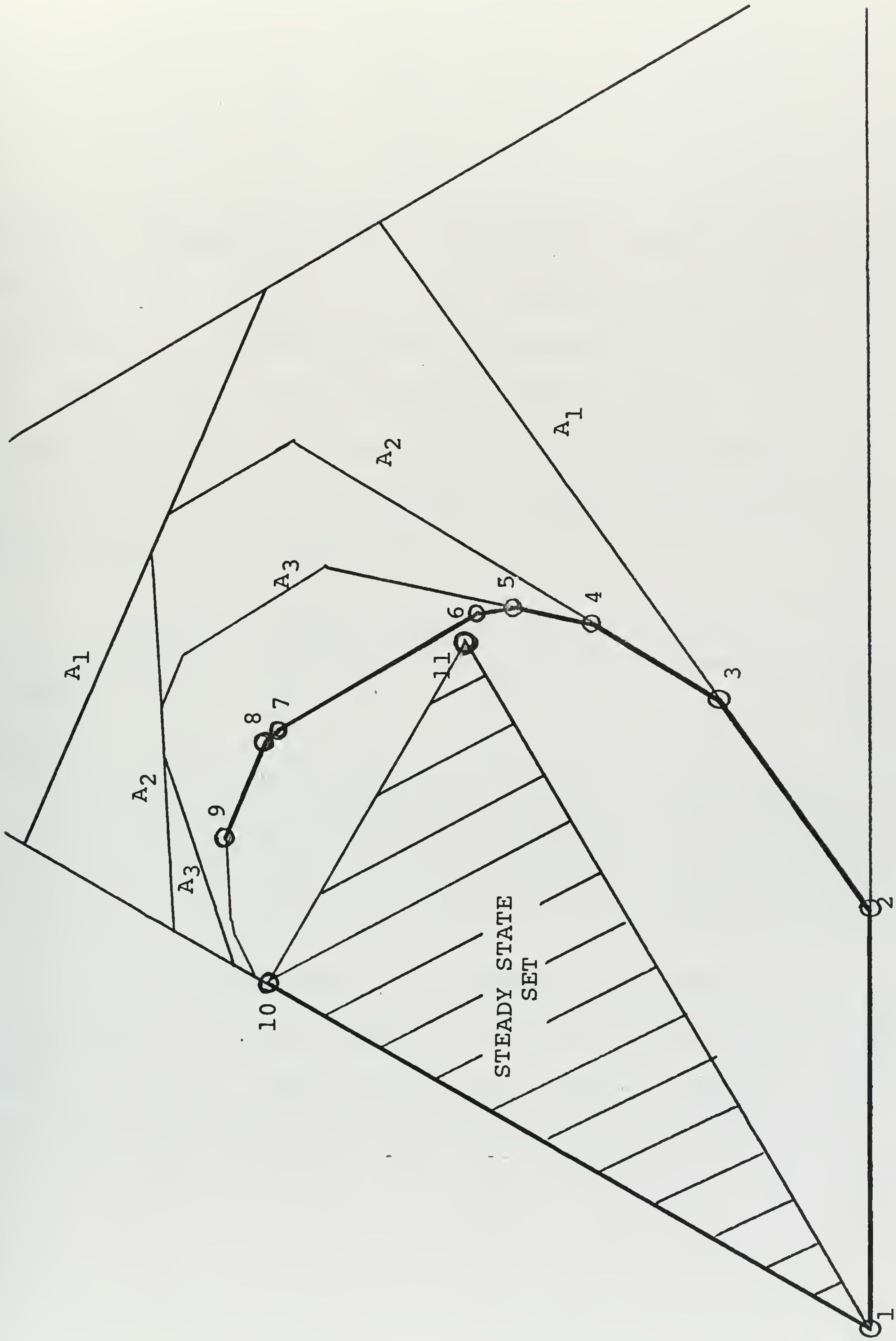

FIGURE 1. ATTAINABLE SETS ON BARYCENTRIC COORDINATES

$$P = \begin{bmatrix} .5 & .4 & 0 \\ 0 & .6 & .3 \\ 0 & 0 & .7 \end{bmatrix} .$$

Table I contains the coordinates of the numbered points on Figure 1. The heavy line in the figure represents the known portion of the boundary of the Containment set after calculating A_7. The portion of the boundary between points 9 and 10 changed very little between A_6 and A_7 so the boundary of A_7 is probably a good approximation of the boundary of the Containment set in this case.

TABLE I: POINTS ASSOCIATED WITH FIGURE 1.

Point / Rank	1	2	3	4	5	6	7	8	9	10	11
1	0	.3	.39	.393	.3723	.35289	.18615	.176445	.0882225	0	1/3
2	0	0	.12	.228	.2940	.32532	.49206	.500706	.5355714	.5	1/3
3	1	.7	.49	.379	.3337	.32179	.32179	.322849	.3762061	.5	1/3

It is not clear how this graphical projection method can be extended to higher dimensions to obtain the extreme points. Nor is it clear, based on the above example, whether or not the Containment set has a finite number of extreme points.

VI. THE DEVELOPMENT OF SEVERAL TESTS

Since it is not feasible to analytically determine the extreme points of all of the sets, we must devise some method to answer our original questions. The answers to those questions depend on knowing the extreme points of all of the sets. To provide partial answers to the questions, we develop test procedures to determine if a given vector is an element of a particular set, and to determine if a given vector can be returned to in m steps or reached from another specified vector in m steps. We know that a given vector cannot be returned to at any time if there exists an i such that the test vector is not in A_i.

Each transition from one set to another requires that a certain set of inequalities be satisfied. When we consider the system of inequalities that account for all of the transitions, we need only test for feasibility. If no feasible solution exists, then the test vector fails the test. Several specific tests are developed in the next sections.

A. A TEST FOR B_m

In this section, we develop a test to determine if a given rank distribution is an element of the set B_m. Recall that $B_m = \{z_m;\ z_m \geq z_{m-1} \cdot P,\ z_m \in F,\ z_{m-1} \in B_{m-1}\}$ for m = 1,2,... and that B_m is the set of rank distributions which is reachable in m steps from the Steady State set. Row vectors have been used in all developments. To simplify notation, the test inequalities will be specified in tableau form and vectors

will be column vectors. Transposition is indicated for the promotion matrix where it appears.

We now write the sequence of inequalities which must be satisfied if the vector X is in the set B_m as

$$
\begin{array}{cccccccl}
z_0 & z_1 & z_2 & \cdots & z_{m-2} & z_{m-1} & & \\
(P^T-I) & & & & & & \leq 0 & \\
P^T & -I & & & & & \leq 0 & \\
 & P^T & -I & & & & \leq 0 & \\
 & & & \ddots & & & \vdots & \\
 & & & & P^T & -I & \leq 0 & \\
 & & & & & P^T & \leq X^T & \qquad (4) \\
e & & & & & & = 1 & \\
 & e & & & & & = 1 & \\
 & & & \ddots & & & \vdots & \\
 & & & & e & & = 1 & \\
 & & & & & e & = 1 &
\end{array}
$$

The constraint matrix appears deceptively smaller than it is. Since each z_i is an n vector, the dimensions of the matrix are mn + m + n rows and mn columns. Even though the constraint matrix is very sparse, we were unable to apply any algorithm which would process the problem more efficiently than the Simplex algorithm.

It is clear that a feasible solution to (4) is simply a Phase I Simplex solution. The method which we use to obtain this solution is to add an artificial variable to each of the feasibility constraints, the last m rows. We then minimize the sum of the artificial variables. A feasible solution to the new system, (4) plus the artificial variables, always exists. The result of the test is determined by the values of the artificial variables. If all of the artificial variables are equal to zero, then (4) has a feasible solution and we conclude that the test vector, X, is an element of B_m. If any of the artificial variables are not zero, then a feasible solution to (4) does not exist and we conclude that X is not in B_m.

B. A TEST FOR A_m

In this section, we develop a test to determine if a given rank distribution is an element of A_m, i.e. if it can be reached in m steps from some point in F. Recall that $A_m = \{z_m; z_m \geq z_{m-1} \cdot P, z_m \in F, z_{m-1} \in A_{m-1}\}$ for m = 1,2,... We write the set of inequalities as

$$
\begin{array}{cccccccl}
z_0 & z_1 & z_2 & \cdots & z_{m-2} & z_{m-1} & & \\
P^T & -I & & & & & \leq & 0 \\
 & P^T & -I & & & & \leq & 0 \\
 & & & \ddots & & & & \vdots \\
 & & & & P^T & -I & \leq & 0 \\
 & & & & & P^T & \leq & X^T \\
e & & & & & & = & 1 \\
 & e & & & & & = & 1 \\
 & & & \ddots & & & & \vdots \\
 & & & & e & & = & 1 \\
 & & & & & e & = & 1
\end{array}
\qquad (5)
$$

The constraint matrix has mn + m rows and mn columns. We now test (5) for a feasible solution in the same manner as the test for B_m. The results of the test are interpreted in the same manner as those for B_m. Note that the constraint matrix in (5) is the same as that in (4) without the first n rows.

C. A TEST FOR RETURN IN m STEPS

In this section, we develop a test to determine whether or not a given rank distribution, X, can be returned to in m steps. If X is an element of the Steady State set, we know that it can be returned to in one step, i.e. it can be maintained forever. So we are interested in a test for m

greater than one. Since there are m transitions, there must be m-1 intermediate vectors because the initial and terminal vectors are both X. The first intermediate vector, z_1, must satisfy $z_1 \geq X \cdot P$. The second intermediate vector, z_2, must satisfy $z_2 \geq z_1 \cdot P$. This process continues until the last intermediate vector, z_{m-1}, must satisfy $z_{m-1} \cdot P \leq X$. We write the sequence of inequalities in tableau form as

$$
\begin{array}{cccccl}
z_1 & z_2 & \cdots & z_{m-2} & z_{m-1} & \\
I & & & & & \geq P^T \cdot X^T \\
P^T & -I & & & & \leq 0 \\
 & P^T & & & & \leq 0 \\
 & & \ddots & & & \;\vdots \\
 & & & P^T & -I & \leq 0 \\
 & & & & P^T & \leq X^T \\
e & & & & & = 1 \\
 & e & & & & = 1 \\
 & & \ddots & & & \;\vdots \\
 & & & e & & = 1 \\
 & & & & e & = 1
\end{array}
\qquad (6)
$$

Again we notice that there is a great similarity between the form of this tableau and those for the tests for A_m and

B_m. The first n inequalities in (6) are different from those in (4), as well as there being one less intermediate vector. The constraint matrix in (6) has mn + m + n rows and mn - n columns.

As with the tests for A_m and B_m, any method of finding a feasible solution to (6) is acceptable. One method would be to add artificial variables not only to the last m rows of (6), but also to the first n rows, then minimizing the sum of the artificial variables. If all of the artificial variables are zero, then we conclude that the rank distribution, X, can be returned to in m steps. The method used in section VII involved adding artificial variables only to the last m rows of (6), then minimizing the sum of the artificial variables. This system may not always have a feasible solution. The test requires that if the system has a feasible solution, then if all of the artificial variables are zero, we conclude that X can be returned to in m steps.

D. A TEST FOR CHANGING DISTRIBUTIONS IN m STEPS

We now develop a test to determine, given some distribution vector, X_1, whether or not we can reach another specified distribution vector, X_2, in m steps. This is nearly identical to the problem discussed in the above section, i.e. we have a fixed initial vector and a fixed terminal vector. The constraint matrix is the same as (6) but now the right hand side is $[P^T \cdot X_1^T, 0, \ldots, 0, X_2^T, 1, \ldots, 1]$. The test procedure is then the same as the test for return in m steps.

E. A TEST FOR EXTREME POINTS OF THE CONTAINMENT SET

The similarity between the tests for A_m and B_m noted in sections A and B above permits us to develop a procedure with which we can determine extreme points of the Containment set, C.

Let $G_1 = [(I-P^T), 0, \ldots, 0]$ be an n by mn matrix and let

$$G_2 = \begin{bmatrix} P^T & -I & & & \\ & P^T & -I & & \\ & & \ddots & \ddots & \\ & & & P^T & -I \\ & & & & P^T \\ e & & & & \\ -e & & & & \\ & e & & & \\ & -e & & & \\ & & \ddots & & \\ & & & e & \\ & & & -e & \\ & & & & e \\ & & & & -e \end{bmatrix}$$

be an mn + 2m by mn matrix. Now let $X_2 = [0,\ldots,0,X^T,1,\ldots,1]$ be an mn + 2m by 1 column vector and $Z = [z_0^T,\ldots,z_{m-1}^T]$ be an mn by 1 column vector. Then the test for A_m reduces to finding

a feasible solution to

$$G_2 \cdot Z \leq X_2 \ . \tag{7}$$

If there is a Z^* which satisfies (7) then the test for B_m would seem to result in testing to see if $G_1 \cdot Z^* \geq 0$. However, there is no guarantee that Z^* is unique and there could exist a Z' which solves (7) and also solves $G_1 \cdot Z' \geq 0$ for the same test vector X. So this would be a sufficient test for inclusion in B_m, but not necessary. The only necessary and sufficient test is that described in section III.A.

Now let $H_1 = [(I-P^T), 0, \ldots, 0]$ be an n by mn + n matrix and let

$$H_2 = \begin{bmatrix} P^T & -I & & & & \\ & P^T & -I & & & \\ & & \ddots & & -I & \\ & & & \ddots & P^T & -I \\ e & & & & & \\ -e & & & & & \\ & e & & & & \\ & -e & & & & \\ & & & & e & \\ & & & & -e & \\ & & & & & e \\ & & & & & -e \end{bmatrix}$$

be an mn + n + 2m by mn + n matrix. Let $Z' = [z_0^T,\ldots,z_m^T]$ be an mn + n by 1 column vector and let $X_2' = [0,\ldots,0,1,\ldots,1]$ be an mn + n + 2m by 1 column vector. This amounts to letting the test vector, X_i^T become an unknown, z_m^T. Then all of the basic feasible solutions to $H_2 \cdot Z' \leq X_2'$ are extreme points of A_m. If a point is an extreme point of A_m and B_m simultaneously, then it is an extreme point of the Containment set, C. Thus we need only test each basic feasible solution to see if $H_1 \cdot Z' \geq 0$. If it is, then Z' is an extreme point of C.

Due to the computational difficulties associated with this test, it is presented for theoretical interest only.

VII. A CASE STUDY

The tests developed in section VI above were applied to faculty distributions in the College of Engineering at the University of California at Berkeley based on data from 1960 to 1968. The bulk of the data is in Branchflower [1970] and only that which is relevant is reproduced here.

We used the Mathematical Programming System on the IBM 360/67 computer at the W. R. CHURCH Computer Center, Naval Postgraduate School to conduct most of the tests. This is really a trivial exercise once the data cards are prepared for the control programs. One FORTRAN IV program was written to generate the data cards for the test for B_m. The same cards were used to test for A_m after removing the cards pertaining to the first n rows of the constraint matrix. Another FORTRAN IV program was written which was used to generate data cards for testing return in m steps. Again, hand modification of the right hand side permitted the same cards to be used to test for changing distributions in m steps.

One other FORTRAN IV program was written which was used to determine the extreme points of the steady state set. It was also used to test to determine if a given vector was an element of the Steady State set.

All of the tests were verified for accuracy by using a test case, namely the three state promotion matrix in section V and points which were in known sets from Figure 1.

STATE I \ J	1	2	3	4	5	6	7	8	9	10	11	12	13
1	.28	.56	.04	.04									
2		.41	.47	.07									
3			.44	.37	.14								
4				.38	.52	.04							
5					.47	.43	.10						
6						.57	.41	.01					
7							.64	.34					
8								.67	.30	.01			
9									.69	.25	.02		
10										.71	.24	.01	
11											.73	.25	.01
12												.77	.20
13													.97

FIGURE 2. PROMOTION MATRIX FROM 1960-1968 DATA

TABLE II. TEST VECTORS

STATE \ VECTOR	X_1(1965)	X_2(1966)	X_3(1967)	X_4(1968)
1	.0161427	.0154639	.0	.0047619
2	.0802139	.0567010	.0628019	.0190476
3	.0534759	.0979381	.0821256	.0809523
4	.0374331	.0257731	.0628019	.0761904
5	.0534759	.0618556	.0483091	.0523809
6	.0481283	.0463917	.0628019	.0666666
7	.1390374	.0979381	.0821256	.0952380
8	.1176470	.1494845	.1400966	.1047619
9	.1122994	.1134020	.1159420	.1238095
10	.1390374	.1030927	.1062801	.0904761
11	.0962566	.1185567	.1014492	.1047619
12	.0855614	.0670103	.0676328	.1047619
13	.0213903	.0463917	.0717328	.0761904

For the case study, we used the thirteen active states of the faculty distribution and the 1960-1968 promotion matrix from Branchflower [1970]. This is reproduced as Figure 2. The actual distributions of faculty for 1965-1968 were normalized and used as the four test vectors. These are included in Table II.

The test vectors were tested for being elements of A_m and B_m, m = 1,2,...,7, and were tested to determine if they could be returned to in m steps for m = 2,3,...,8. The resulting location of the vectors of rank distributions on the fundamental simplex is indicated in Table III.

TABLE III. TEST RESULTS

X_1	X_2	X_3	X_4
$X_1 \in A_2$		$X_3 \in A_2$	$X_4 \in A_2$
$X_1 \notin A_3$	$X_2 \notin A_1$	$X_3 \notin A_3$	$X_4 \notin A_3$

Recall that if $X_i \in A_j$ then after one step, $X_{i+1} \geq X_i \cdot P$, X_{i+1} is at least an element of A_{j+1}. The results in Table III are inconsistent with this theoretical result and require an explanation. There are two reasons why this could have occured. The first reason is that the Promotion matrix was an average over an eight year period and did not exactly represent the true Promotion matrix for each year. The second reason is that the faculty size did not remain constant, but in fact was increasing. An expanding organization can appoint personnel

to replace the losses and then hire the additional number for expansion into any state. When P is upper triangular, the extreme points of successive A_m may tend to move away from the lower ranks. For a three state example, consider an initial rank distribution of (1, 0, 0) with

$$P = \begin{bmatrix} .5 & .4 & 0 \\ 0 & .6 & .3 \\ 0 & 0 & .7 \end{bmatrix}$$

then after one transition, the rank distribution before new appointments is (.5, .4, 0). Say the number of persons in the organization is constant at 100, then the distribution of persons in the ranks is (50, 40, 0) with a loss of 10 to replace. Appointments could result in the distributions (60, 40, 0), (50, 50, 0), or (50, 40, 10) or any linear combination of them. If the organization expands by 10 persons, then the distributions could be (70, 40, 0), (50, 60, 0), or (50, 40, 20) or any linear combination of them. If we normalize, then these expanded distributions are outside A_1 for the constant size case. The result is that the possible distributions for the expanding size model can be outside of the normal set of distributions obtained for the constant size model.

We then aggregated several ranks in the model to determine if the aggregated model behaved in the same way. The first four ranks are Assistant Professors, the next three ranks are

Associate Professors, and the last six ranks are Full Professors. The resulting Promotion matrix is

$$P = \begin{bmatrix} .7649 & .1752 & 0 \\ 0 & .8030 & .1841 \\ 0 & 0 & .9707 \end{bmatrix}$$

The normalized rank distributions for 1965-1968 are listed in Table IV.

TABLE IV. TEST VECTORS: AGGREGATED MODEL

RANK \ VECTOR	X_1(1965)	X_2(1966)	X_3(1967)	X_4(1968)
Asst. Prof.	.187166	.195876	.207729	.180952
Assoc. Prof.	.240642	.206186	.193237	.214286
Full Prof.	.572192	.597938	.599034	.604762

These vectors were tested to determine their location on the fundamental simplex. The results are shown in Table V.

TABLE V. TEST RESULTS: AGGREGATED MODEL

X_1	X_2	X_3	X_4
$X_1 \in A_9$	$X_2 \in A_7$	$X_3 \in A_6$	$X_4 \in A_9$
$X_1 \notin A_{10}$	$X_2 \notin A_8$	$X_3 \notin A_7$	$X_4 \notin A_{10}$

These results using the aggregated model are similar to the results of the thirteen state model. We noted above that as faculty size increases, the rank distribution may move to a superset of the set containing the original distribution. In fact, this happens in 1966 for an increase of seven faculty members (3.7%), and also in 1967 for an increase of thirteen faculty members. In 1968, there was an increase of three faculty members, but the rank distribution moved to a subset of the set containing the 1967 rank distribution.

VIII. CONCLUSIONS AND RECOMMENDATIONS

We have established several properties of several sets of rank distribution in an hierarchical organization which have not been previously examined. In addition, based on these properties, we developed several tests to determine if a specified rank distribution is an element of certain sets. The tests were applied to data from a case study of the College of Engineering faculty at the University of California, Berkeley.

Several questions remain unanswered and are possible areas for further study. Is there some neighborhood of a given rank distribution which can be maintained? Is there an efficient algorithm which will solve the test inequalities other than the Simplex method? Is there a method of reducing the number of basic feasible solutions which must be found to determine the extreme points of the Containment set? Does the Containment set have a finite number of extreme points?

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

* * * * DATA GENERATOR - TEST FOR RETURN IN M STEPS * * * *

* * * * * * ROBERT L. ARMACOST - - - 1 JULY 1970 * * * * * *

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *


* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
     THIS PROGRAM WILL PUNCH OUT A SET OF DATA CARDS FOR A
     LINEAR PROGRAM USING THE MPS PACKAGE ON THE IBM 360/67.
     THE BASIC INPUT TO THE PROGRAM IS A SQUARE PROMOTION
     MATRIX OF SIZE N, AND K VECTORS WHICH ARE TO BE TESTED
     TO DETERMINE IF THE VECTOR CAN BE RETURNED TO IN M
     STEPS.  THE PROGRAM WILL GENERATE DATA DECKS FOR A
     SERIES OF TESTS TO RETURN IN M STEPS.  MSTART IS THE
     STARTING INDEX FOR THE NUMBER OF STEPS TO RETURN, AND
     MSTOP IS THE ENDING INDEX.  THESE INDICES ARE READ IN
     AS DATA.  THE FIRST VECTOR TO BE TESTED IS CALLED
     "TEST".  EACH SUCCEEDING VECTOR IS CALLED "REVI" WHERE
     I IS THE NUMBER OF THE REVISION OF THE RIGHT HAND SIDE.
     THE VARIABLE IR IS THE NUMBER OF REVISIONS IN THE
     CONTROL PROGRAM.  THE ARRAYS SHOULD BE DIMENSIONED IN
     THE FOLLOWING WAY -
          AX((MSTOP-1)*(N+1) + N, (MSTOP-1)*(N+1))
          KEY - SAME AS AX
          X(N,IR+1)
          P(N,N)
          PT(N,N)
          PX(N,N)
          UNI(N,N).

     THE PROGRAM WILL ACCOMODATE A PROMOTION MATRIX WITH
     N = 99 AND WITH MSTOP < 100.

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
     THERE IS A CERTAIN AMOUNT OF REPETITION BETWEEN THIS
     PROGRAM AND THE FOLLOWING PROGRAM FOR A TEST FOR A
     REACHABLE SET IN M STEPS.  TO MINIMIZE DOCUMENTATION,
     CERTAIN MODULES IN THIS PROGRAM NEED ONLY BE REPLACED
     BY OTHER MODULES TO OBTAIN THE NEW PROGRAM.  THESE
     MODULES ARE MARKED WITH A NUMBER IN THE LEFT HAND
     COLUMN WHICH MEANS THEY ARE TO BE REPLACED BY THE
     STATEMENTS WITH THE CORRESPONDING NUMBER IN THE NEXT
     SECTION TO OBTAIN THE NEW PROGRAM.  IN ADDITION, THERE
     ARE SEVERAL STATEMENTS WHICH NEED ONLY BE REMOVED.
     THESE ARE MARKED WITH AN R IN THE LEFT HAND COLUMN.
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

      INTEGER*2 KEY(111,98)
      DIMENSION AX(111,98), X(13,4), P(13,13), PT(13,13),
     1UNI(13,13), PX(13,13)

     READ INPUT DATA

1     READ(5,600) MSTART, MSTOP, N, IR
 600  FORMAT(4I5)
```

```
       IR1 = IR + 1
       READ(5,601) ((P(I,J),J=1,N),I=1,N)
 601   FORMAT(8F10.0)
       READ(5,601) ((X(I,J),I=1,N),J=1,IR1)
R      MBEG = MSTART - 1
R      MEND = MSTOP - 1
       DO 999 M = MBEG,MEND

      INITIALIZE

       ITEST = 1
       JLIM = M*N + M
       ILIM = JLIM + N
       DO 10 J=1,JLIM
       DO 10 I =1,ILIM
       AX(I,J) = 0.
 10    KEY(I,J) = 0

2     CALCULATE THE TRANSPOSE OF THE PROMOTION MATRIX, PT,
2     AND LOAD THE IDENTITY MATRIX UNI
2
2      DO 20 I=1,N
2      DO 20 J=1,N
2      UNI(I,J) = 0.
2      PT(I,J) = P(J,I)
2      IF(I.EQ.J) UNI(I,J) = 1.
2   20 CONTINUE

R     CALCULATE THE RHS FOR THE FIRST N ROWS
R
R      DO 1 J = 1,IR1
R      DO 2 I = 1,N
R      SUM = 0.
R      DO 3 K = 1,N
R    3 SUM = SUM + PT(I,K) * X(K,J)
R    2 PX(I,J) = SUM
R    1 CONTINUE

3     INSERT UNI IN THE FIRST N ROWS AND COLUMNS OF THE
3     CONSTRAINT MATRIX, AX
3
3      DO 30 I=1,N
3      DO 30 J=1,N
3      AX(I,J) = UNI(I,J)
3   30 IF(AX(I,J).NE.0) KEY(I,J) = 1

      INSERT PT AND -I IN THE APPROPRIATE COLUMNS OF THE NEXT
4     (M-2)*N ROWS OF AX

       LASTC = N
       NXTR = N+1
       NXTC = N+1
       JLASTC = 1
       LASTR = 2*N
       JNXTC = 2*N
       MOD = 1
 5     IF(MOD.EQ.M) GO TO 100
       DO 50 I = NXTR,LASTR
       DO 40 J = JLASTC,LASTC
       AX(I,J) = PT(I-MOD*N,J-(MOD-1)*N)
 40    IF(AX(I,J).NE.0) KEY(I,J) = 1
       DO 50 J = NXTC,JNXTC
       IF(I .EQ.J) AX(I,J) = - 1.0
 50    IF(AX(I,J).EQ.-1.0) KEY(I,J) = 1
       NXTC = NXTC + N
       NXTR = NXTR + N
       LASTC = JNXTC
       JLASTC = NXTC - N
       LASTR = NXTR + N - 1
       JNXTC = LASTC + N
       MOD = MOD + 1
       GO TO 5
```

```
C     INSERT PT IN THE CORRECT COLUMNS OF THE NEXT N ROWS OF
C     AX
C
  100 DO 60 I = NXTR,LASTR
      DO 60 J = JLASTC,LASTC
      AX(I,J) = PT(I-MOD*N,J-(MOD-1)*N)
   60 IF(AX(I,J).NE.0) KEY(I,J) = 1
C
C     INSERT 1'S IN THE CORRECT COLUMNS OF THE LAST M ROWS OF
C     AX WHICH CORRESPOND TO THE FEASIBILITY CONSTRAINT
C
      MOD = MOD + 1
      IM = LASTR + 1
      IMM = LASTR + M
      DO 70 I=IM,IMM
      ICOL = (I-IM)*N + 1
      JCOL = ICOL - 1 + N
      DO 65 J = ICOL,JCOL
      AX(I,J) = 1.
   65 KEY(I,J) = 1
      K = M*N + I - LASTR
      AX(I,K) = 1.
      KEY(I,K) = 1
   70 CONTINUE
C
C     PUNCH OUT THE DATA SET NAME AND ROWS CONSTRAINTS CARDS
C
      WRITE(6,700)
  700 FORMAT('NAME',T15,'TEST')
      WRITE(6,701)
  701 FORMAT('ROWS')
      WRITE(6,702)
  702 FORMAT(T2,'N  OBJF')
      DO 200 J=1,N
      IF(J.GT.9) GO TO 350
      WRITE(6,703) J
  703 FORMAT(T2,'G  AO,'I1)
      GO TO 200
  350 WRITE(6,753) J
  753 FORMAT(T2,'G  AO,',I2)
  200 CONTINUE
      DO 201 I =1,M
      DO 201 J =1,N
      IF(I.LE.9.AND.J.LE.9) GO TO 351
      IF(I.LE.9.AND.J.GT.9) GO TO 352
      IF(I.GT.9.AND.J.LE.9) GO TO 353
      IF(I.GT.9.AND.J.GT.9) GO TO 354
  351 WRITE(6,741) I,J
  741 FORMAT(T2,'L  A',I1,',',I1)
      GO TO 201
  352 WRITE(6,742) I,J
  742 FORMAT(T2,'L  A',I1,',',I2)
      GO TO 201
  353 WRITE(6,743) I,J
  743 FORMAT(T2,'L  A',I2,',',I1)
      GO TO 201
  354 WRITE(6,744) I,J
  744 FORMAT(T2,'L  A',I2,',',I2)
  201 CONTINUE
      DO 202 I = 1,M
      IF(I.GT.9) GO TO 320
      WRITE(6,705) I
  705 FORMAT(T2,'E  AR',I1)
      GO TO 202
  320 WRITE(6,720) I
  720 FORMAT(T2,'E  AR',I2)
  202 CONTINUE
C
C     PUNCH OUT THE COLUMNS DATA CARDS
C
      WRITE(6,706)
```

```
706   FORMAT('COLUMNS')
      JX = 0
      JEND = M*N
      DO 300 J = 1,JEND
      JX = JX + 1
      DO 302 K = 1,M
      IF(J.LE.K*N.AND.J.GT.(K-1)*N) IX = K-1
      IF(J.EQ.IX*N+1) JX = 1
302   CONTINUE
      DO 301 I = 1,ILIM
      IF(KEY(I,J).NE.1) GO TO 301
      M2 = M + 2
      DO 303 L = 1,M2
      IF(I.LE.L*N.AND.I.GT.(L-1)*N) IA = L-1
303   CONTINUE
      JA = I - IA*N
      IF(I.GT.(M+1)*N) JA = I - (M+1)*N
      IF(IX.LE.9.AND.JX.LE.9.AND.IA.LE.9.AND.JA.LE.9) GO TO
     1360
      IF(IX.LE.9.AND.JX.GT.9.AND.IA.LE.9.AND.JA.LE.9) GO TO
     1361
      IF(IX.LE.9.AND.JX.GT.9.AND.IA.LE.9.AND.JA.GT.9) GO TO
     1362
      IF(IX.LE.9.AND.JX.GT.9.AND.IA.GT.9.AND.JA.LE.9) GO TO
     1363
      IF(IX.LE.9.AND.JX.GT.9.AND.IA.GT.9.AND.JA.GT.9) GO TO
     1364
      IF(IX.GT.9.AND.JX.GT.9.AND.IA.LE.9.AND.JA.GT.9) GO TO
     1365
      IF(IX.GT.9.AND.JX.GT.9.AND.IA.GT.9.AND.JA.LE.9) GO TO
     1366
      IF(IX.GT.9.AND.JX.GT.9.AND.IA.GT.9.AND.JA.GT.9) GO TO
     1367
      IF(IX.LE.9.AND.JX.LE.9.AND.IA.LE.9.AND.JA.GT.9) GO TO
     1368
      IF(IX.GT.9.AND.JX.LE.9.AND.IA.GT.9.AND.JA.LE.9) GO TO
     1369
      IF(IX.LE.9.AND.JX.LE.9.AND.IA.GT.9.AND.JA.GT.9) GO TO
     1370
      IF(IX.GT.9.AND.JX.LE.9.AND.IA.GT.9.AND.JA.GT.9) GO TO
     1371
      IF(IX.LE.9.AND.JX.LE.9.AND.IA.GT.9.AND.JA.LE.9) GO TO
     1372
      IF(IX.GT.9.AND.JX.LE.9.AND.IA.LE.9.AND.JA.GT.9) GO TO
     1373
      IF(IX.GT.9.AND.JX.GT.9.AND.IA.LE.9.AND.JA.LE.9) GO TO
     1374
      IF(IX.GT.9.AND.JX.LE.9.AND.IA.LE.9.AND.JA.LE.9) GO TO
     1375
360   IF(I.GT.(M+1)*N) GO TO 380
      WRITE(6,760) IX,JX,IA,JA,AX(I,J)
760   FORMAT(T5,'X',I1,',',I1,T15,'A',I1,',',I1,T25,F10.7)
      GO TO 301
380   WRITE(6,780) IX,JX,JA,AX(I,J)
780   FORMAT(T5,'X',I1,',',I1,T15,'AR',I1,T25,F10.7)
      GO TO 301
361   IF(I.GT.(M+1)*N) GO TO 381
      WRITE(6,761) IX,JX,IA,JA,AX(I,J)
761   FORMAT(T5,'X',I1,',',I2,T15,'A',I1,',',I1,T25,F10.7)
      GO TO 301
381   WRITE(6,781) IX,JX,JA,AX(I,J)
781   FORMAT(T5,'X',I1,',',I2,T15,'AR',I1,T25,F10.7)
      GO TO 301
362   IF(I.GT.(M+1)*N) GO TO 382
      WRITE(6,762) IX,JX,IA,JA,AX(I,J)
762   FORMAT(T5,'X',I1,',',I2,T15,'A',I1,',',I2,T25,F10.7)
      GO TO 301
382   WRITE(6,782) IX,JX,JA,AX(I,J)
782   FORMAT(T5,'X',I1,',',I2,T15,'AR',I2,T25,F10.7)
      GO TO 301
363   IF(I.GT.(M+1)*N) GO TO 383
      WRITE(6,763) IX,JX,IA,JA,AX(I,J)
```

```
763   FORMAT(T5,'X',I1,',',I2,T15,'A',I2,',',I1,T25,F10.7)
      GO TO 301
383   WRITE(6,783) IX,JX,JA,AX(I,J)
783   FORMAT(T5,'X',I1,',',I2,T15,'AR',I1,T25,F10.7)
      GO TO 301
364   IF(I.GT.(M+1)*N) GO TO 384
      WRITE(6,764) IX,JX,IA,JA,AX(I,J)
764   FORMAT(T5,'X',I1,',',I2,T15,'A',I2,',',I2,T25,F10.7)
      GO TO 301
384   WRITE(6,784) IX,JX,JA,AX(I,J)
784   FORMAT(T5,'X',I1,',',I2,T15,'AR',I2,T25,F10.7)
      GO TO 301
365   IF(I.GT.(M+1)*N) GO TO 385
      WRITE(6,765) IX,JX,IA,JA,AX(I,J)
765   FORMAT(T5,'X',I2,',',I2,T15,'A',I1,',',I2,T25,F10.7)
      GO TO 301
385   WRITE(6,785) IX,JX,JA,AX(I,J)
785   FORMAT(T5,'X',I2,',',I2,T15,'AR',I2,T25,F10.7)
      GO TO 301
366   IF(I.GT.(M+1)*N) GO TO 386
      WRITE(6,766) IX,JX,IA,JA,AX(I,J)
766   FORMAT(T5,'X',I2,',',I2,T15,'A',I2,',',I1,T25,F10.7)
      GO TO 301
386   WRITE(6,786) IX,JX,JA,AX(I,J)
786   FORMAT(T5,'X',I2,',',I2,T15,'AR',I1,T25,F10.7)
      GO TO 301
367   IF(I.GT.(M+1)*N) GO TO 387
      WRITE(6,767) IX,JX,IA,JA,AX(I,J)
767   FORMAT(T5,'X',I2,',',I2,T15,'A',I2,',',I2,T25,F10.7)
      GO TO 301
 387   WRITE(6,787) IX,JX,JA,AX(I,J)
787   FORMAT(T5,'X',I2,',',I2,T15,'AR',I2,T25,F10.7)
      GO TO 301
368   IF(I.GT.(M+1)*N) GO TO 388
      WRITE(6,768) IX,JX,IA,JA,AX(I,J)
768   FORMAT(T5,'X',I1,',',I1,T15,'A',I1,',',I2,T25,F10.7)
      GO TO 301
388   WRITE(6,788) IX,JX,JA,AX(I,J)
788   FORMAT(T5,'X',I1,',',I1,T15,'AR',I2,T25,F10.7)
      GO TO 301
369   IF(I.GT.(M+1)*N) GO TO 389
      WRITE(6,769) IX,JX,IA,JA,AX(I,J)
769   FORMAT(T5,'X',I2,',',I1,T15,'A',I2,',',I1,T25,F10.7)
      GO TO 301
389   WRITE(6,789) IX,JX,JA,AX(I,J)
789   FORMAT(T5,'X',I2,',',I1,T15,'AR',I1,T25,F10.7)
      GO TO 301
370   IF(I.GT.(M+1)*N) GO TO 390
      WRITE(6,770) IX,JX,IA,JA,AX(I,J)
770   FORMAT(T5,'X',I1,',',I1,T15,'A',I2,',',I2,T25,F10.7)
      GO TO 301
390   WRITE(6,790) IX,JX,JA,AX(I,J)
790   FORMAT(T5,'X',I1,',',I1,T15,'AR',I2,T25,F10.7)
      GO TO 301
371   IF(I.GT.(M+1)*N) GO TO 391
      WRITE(6,771) IX,JX,IA,JA,AX(I,J)
771   FORMAT(T5,'X',I2,',',I1,T15,'A',I2,',',I2,T25,F10.7)
      GO TO 301
391   WRITE(6,791) IX,JX,JA,AX(I,J)
791   FORMAT(T5,'X',I2,',',I1,T15,'AR',I2,T25,F10.7)
      GO TO 301
372   IF(I.GT.(M+1)*N) GO TO 392
      WRITE(6,772) IX,JX,IA,JA,AX(I,J)
772   FORMAT(T5,'X',I1,',',I1,T15,'A',I2,',',I1,T25,F10.7)
      GO TO 301
392   WRITE(6,792) IX,JX,JA,AX(I,J)
792   FORMAT(T5,'X',I1,',',I1,T15,'AR',I1,T25,F10.7)
      GO TO 301
373   IF(I.GT.(M+1)*N) GO TO 393
      WRITE(6,773) IX,JX,IA,JA,AX(I,J)
773   FORMAT(T5,'X',I2,',',I1,T15,'A',I1,',',I2,T25,F10.7)
      GO TO 301
```

```
393   WRITE(6,793) IX,JX,JA,AX(I,J)
793   FORMAT(T5,'X',I2,',',I1,T15,'AR',I2,T25,F10.7)
      GO TO 301
374   IF(I.GT.(M+1)*N) GO TO 394
      WRITE(6,774) IX,JX,IA,JA,AX(I,J)
774   FORMAT(T5,'X',I2,',',I2,T15,'A',I1,',',I1,T25,F10.7)
      GO TO 301
394   WRITE(6,794) IX,JX,JA,AX(I,J)
794   FORMAT(T5,'X',I2,',',I2,T15,'AR',I1,T25,F10.7)
      GO TO 301
375   IF(I.GT.(M+1)*N) GO TO 395
      WRITE(6,775) IX,JX,IA,JA,AX(I,J)
775   FORMAT(T5,'X',I2,',',I1,T15,'A',I1,',',I1,T25,F10.7)
      GO TO 301
395   WRITE(6,795) IX,JX,JA,AX(I,J)
795   FORMAT(T5,'X',I2,',',I1,T15,'AR',I1,T25,F10.7)
301   CONTINUE
      IF(J.LT.JEND) GO TO 300
      DO 305 IZ = 1,M
      JZ = IZ
      IF(IZ.GT.9) GO TO 304
      WRITE(6,709) IZ, JZ
709   FORMAT(T5,'XA',I1,T15,'OBJF',T25,'1.',T40,'AR',I1,T50,
     1'1.')
      GO TO 305
304   WRITE(6,708) IZ, JZ
708   FORMAT(T5,'XA',I2,T15,'OBJF',T25,'1.',T40,'AR',I2,T50,
     1'1.')
305   CONTINUE
300   CONTINUE

    PUNCH OUT THE RIGHT HAND SIDE DATA CARDS

      WRITE(6,710)
710   FORMAT('RHS')
R     DO 325 I = 1,N
R     IF(I.GT.9) GO TO 333
R     WRITE(6,717) I, PX(I,1)
R 717 FORMAT(T5,'B1',T15,'AO,',I1,T25,F10.7)
R     GO TO 325
R 333 WRITE(6,718) I, PX(I,1)
R 718 FORMAT(T5,'B1',T15,'AO,',I2,T25,F10.7)
R 325 CONTINUE
      DO 306 I = 1,N
      IF(M.LE.9.AND.I.LE.9) GO TO 311
      IF(M.LE.9.AND.I.GT.9) GO TO 312
      IF(M.GT.9.AND.I.LE.9) GO TO 313
      IF(M.GT.9.AND.I.GT.9) GO TO 314
311   WRITE(6,711) M, I, X(I,1)
711   FORMAT(T5,'B1',T15,'A',I1,',',I1,T25,F10.7)
      GO TO 306
312   WRITE(6,722) M, I, X(I,1)
722   FORMAT(T5,'B1',T15,'A',I1,',',I2,T25,F10.7)
      GO TO 306
313   WRITE(6,723) M, I, X(I,1)
723   FORMAT(T5,'B1',T15,'A',I2,',',I1,T25,F10.7)
      GO TO 306
314   WRITE(6,724) M, I, X(I,1)
724   FORMAT(T5,'B1',T15,'A',I2,',',I2,T25,F10.7)
306   CONTINUE
      DO 307 I = 1,M
      IF(I.GT.9) GO TO 315
      WRITE(6,712) I
712   FORMAT(T5,'B1',T15,'AR',I1,T25,'1.')
      GO TO 307
315   WRITE(6,725) I
725   FORMAT(T5,'B1',T15,'AR',I2,T25,'1.')
307   CONTINUE
      WRITE(6,713)
713   FORMAT('ENDATA')
801   IF(ITEST.EQ.IR1) GO TO 999
```

```
     PUNCH OUT THE APPROPRIATE RHS REVISION DATA CARDS
      IF(ITEST.GT.9) GO TO 308
      WRITE(6,714) ITEST
  714 FORMAT('NAME',T15,'REV',I1)
      GO TO 309
  308 WRITE(6,715) ITEST
  715 FORMAT('NAME',T15,'REV',I2)
  309 WRITE(6,710)
      WRITE(6,716)
  716 FORMAT(T3,'MODIFY')
      IT1 = ITEST + 1
R     DO 326 I = 1,N
R     IF(I.GT.9) GO TO 334
R     WRITE(6,717) I,PX(I,IT1)
R     GO TO 326
R 334 WRITE(6,718) I, PX(I,IT1)
R 326 CONTINUE
      DO 310 I = 1,N
      IF(M.LE.9.AND.I.LE.9) GO TO 321
      IF(M.LE.9.AND.I.GT.9) GO TO 322
      IF(M.GT.9.AND.I.LE.9) GO TO 323
      IF(M.GT.9.AND.I.GT.9) GO TO 324
  321 WRITE(6,711) M, I, X(I,IT1)
      GO TO 310
  322 WRITE(6,722) M, I, X(I,IT1)
      GO TO 310
  323 WRITE(6,723) M, I, X(I,IT1)
      GO TO 310
  324 WRITE(6,724) M, I, X(I,IT1)
  310 CONTINUE
      WRITE(6,713)
      ITEST = ITEST + 1
      GO TO 801
  999 CONTINUE
      STOP
      END

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
```

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

* * DATA GENERATOR - TEST FOR A REACHABLE SET IN M STEPS * *

* * * * * * ROBERT L. ARMACOST - - - 1 JULY 1970 * * * * * *

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *


* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
     THIS PROGRAM WILL PUNCH OUT A SET OF DATA CARDS FOR A
     LINEAR PROGRAM USING THE MPS PACKAGE ON THE IBM 360/67.
     THE BASIC INPUT TO THE PROGRAM IS A SQUARE PROMOTION
     MATRIX OF SIZE N, AND K VECTORS WHICH ARE TO BE TESTED
     TO DETERMINE IF THEY ARE  ELEMENTS OF THE SET B<M>.
     THE PROGRAM WILL GENERATE DATA DECKS FOR A SERIES OF
     REACHABLE SETS.  MBEG IS THE STARTING INDEX FOR THE
     SUBSCRIPT OF THE TEST.  MEND IS THE ENDING INDEX FOR
     THE SUBSCRIPT OF THE TEST.  THESE INDICES ARE READ IN
     AS DATA.  THE FIRST VECTOR TO BE TESTED IS CALLED
     "TEST".  EACH SUCCEEDING VECTOR IS CALLED "REVI" WHERE
     I IS THE NUMBER OF THE REVISION OF THE RIGHT HAND SIDE.
     THE VARIABLE IR IS THE NUMBER OF REVISIONS IN THE
     CONTROL PROGRAM.  THE ARRAYS SHOULD BE DIMENSIONED IN
     THE FOLLOWING WAY -
          AX((MEND)*(N+1) + N, MEND*(N+1))
          KEY - SAME AS AX
          X(N,IR+1)
          P(N,N)
          PT(N,N)
          PTI(N,N).

     THE PROGRAM WILL ACCOMODATE A PROMOTION MATRIX WITH
     N = 99 AND WITH MEND < 100.

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
     THIS PROGRAM IS CONSTRUCTED BY REMOVING THOSE CARDS
     MARKED WITH AN R IN PROGRAM TO TEST FOR RETURN IN M
     STEPS, AND TO REPLACE THOSE MODULES MARKED WITH NUMBERS
     BY THE MODULES LISTED BELOW WITH THE CORRESPONDING
     NUMBERS.
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

1         READ(5,600) MBEG, MEND, N, IR

2        CALCULATE THE TRANSPOSE OF THE PROMOTION MATRIX, PT,
2        AND CALCULATE I - PT WHERE I IS THE IDENTITY MATRIX
2
2         DO 20 I=1,N
2         DO 20 J=1,N
2         PT(I,J) = P(J,I)
2         PTI(I,J) = -PT(I,J)
2         IF(I.EQ.J) PTI(I,J) = 1. + PTI(I,J)
2      20 CONTINUE

3        INSERT PTI IN THE FIRST N ROWS AND COLUMNS OF THE
3        CONSTRAINT MATRIX, AX
3
```

```
3       DO 30 I=1,N
3       DO 30 J=1,N
3       AX(I,J) = PTI(I,J)
3    30 IF(AX(I,J).NE.0) KEY(I,J) = 1

4      (M-1)*N ROWS OF AX
```

* *

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

* STEADY STATE SET - EXTREME POINTS AND TEST FOR INCLUSION *

* * * * * * ROBERT L. ARMACOST - - - 1 JULY 1970 * * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

     THE MAIN PROGRAM IS DIMENSIONED TO HANDLE A PROMOTION
     MATRIX WITH 15 STATES.  THE PROMOTION MATRIX IS P AND
     ITS INVERSE WHICH HAS BEEN NORMALIZED IS PIN.  THE
     ROWS OF PIN ARE THE EXTREME POINTS OF THE STEADY STATE
     SET.

     THE EXTREME POINTS OF THE STEADY STATE SET ARE USED
     TO DETERMINE IF A TEST POINT IS IN THE STEADY STATE
     SET BY CALCULATING THE NECESSARY COEFFICIENTS TO
     MAKE THE TEST POINT A LINEAR COMBINATION OF THE
     EXTREME POINTS.  IF THE SUM OF THE COEFFICIENTS IS
     GREATER THAN ONE, OR IF ANY COEFFICIENT IS NEGATIVE,
     WE CONCLUDED THAT THE TEST POINT IS NOT IN THE STEADY
     STATE SET.  THE COEFFICIENTS ARE DENOTED ALF, AND THE
     TEST POINTS ARE X.  THE PROGRAM IS SET UP TO ACCEPT
     FIVE SUCH TEST VECTORS WITH FIFTEEN RANKS.

     THE SIZE OF THE PROMOTION MATRIX AND THE LENGTH OF THE
     TEST VECTOR ARE INPUT TO THE PROGRAM IN THE VARIABLE
     NN.  THE NUMBER OF TEST VECTORS IS INPUT IN THE
     VARIABLE KK.

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

      DIMENSION UNI(15,15),P(15,15), PIN(15,15), L(15),
     1M(15), SUM(15), ALF(15), X(15,5)
      READ(5,599) NN, KK
  599 FORMAT(2I5)
      CALL WORK(UNI,P,PIN,L,M,ALF,X,NN,KK,SUM)
      CALL EXIT
      END

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

* * * SUBROUTINE WORK - THIS SUBROUTINE CALCULATES THE * * *
* * * EXTREME POINTS OF THE STEADY STATE SET, M, AND   * * *
* * * TESTS EACH TEST VECTOR FOR INCLUSION IN THAT SET * * *

      SUBROUTINE WORK(UNI,P,PIN,L,M,ALF,X,NN,KK,SUM)
      DIMENSION UNI(NN,NN),P(NN,NN),PIN(NN,NN),L(NN),
     1M(NN), SUM(NN), ALF(NN),X(NN,KK)
      READ(5,600) ((P(I,J),J=1,NN),I=1,NN)
  600 FORMAT(8F10.0)
      READ(5,601) ((X(J,K),J=1,NN),K=1,KK)
  601 FORMAT(8F10.0)

     INITIALIZE

      DO 10 I = 1,NN
      L(I) = 0
      M(I) = 0
```

```
      SUM(I) = 0.
      DO 10 J = 1,NN
      UNI(I,J) = - P(I,J)
      IF(I.EQ.J) UNI(I,J) = 1. + UNI(I,J)
10    CONTINUE
      WRITE(6,699)
C
C     WRITE OUT PROMOTION MATRIX
C
      DO 2 I=1,NN
      WRITE(6,700) (P(I,J),J=1,NN)
      WRITE(6,710)
2     CONTINUE
C
C     CALCULATE THE INVERSE OF P
C
      CALL MINV(UNI,NN,D,L,M)
C
C     CALCULATE THE NORMALIZED INVERSE OF P, PIN
C
      DO 20 I=1,NN
      DO 15 J=1,NN
      SUM(I) = SUM(I) + UNI(I,J)
15    CONTINUE
      DO 25 J=1,NN
      PIN(I,J) = UNI(I,J) / SUM(I)
25    CONTINUE
20    CONTINUE
      WRITE(6,699)
699   FORMAT(1H1)
C
C     WRITE OUT NORMALIZED INVERSE OF P
C
      DO 5 I=1,NN
      WRITE(6,700) (PIN(I,J),J=1,NN)
      WRITE(6,710)
5     CONTINUE
700   FORMAT(15F8.6)
710   FORMAT(1H )
C
C     TEST TO SEE IF TEST VECTOR IS IN THE STEADY STATE SET
C
      DO 130 K=1,KK
      MM=K
      DO 1 I=1,NN
1     ALF(I) = 0.
      ALF(1) = X(1,K) / PIN(1,1)
      TEST = ALF(1)
      JEND = NN-1
      DO 120 J=2,JEND
      IEND = J - 1
      SUMD = 0.
      DO 110 I=1,IEND
110   SUMD = SUMD + ALF(I) * PIN(I,J)
      ALF(J) = (X(J,K) - SUMD) / PIN(J,J)
      TEST = TEST + ALF(J)
      IF(ALF(J).LT.0) GO TO 100
      IF(TEST.GT.1) GO TO 200
120   CONTINUE
      ALF(NN) = 1. - TEST
      IF(ALF(NN).LT.0) GO TO 100
      IF(ALF(NN).LT.X(NN,K)) GO TO 200
130   CONTINUE
      WRITE(6,702)
702   FORMAT(1H1,T20,'THE VECTOR IS IN THE STEADY STATE SET'
     1//T20,'ALF COEFFICIENTS',T45,'TEST VECTOR'//)
      DO 201 LL=1,NN
      WRITE(6,703) ALF(LL), X(LL,MM)
703   FORMAT(T20,F10.8,T45,F10.8)
201   CONTINUE
      IF(K.LT.KK) GO TO 130
      GO TO 999
```

```
100   WRITE(6,704)
704   FORMAT(1H1,T20,'THE VECTOR IS NOT IN THE STEADY STATE'
     1,' SET, ALF IS NEGATIVE'//T20,'ALF COEFFICIENTS',T45,
     2'TEST VECTOR'//)
      DO 202 LL=1,NN
      WRITE(6,703) ALF(LL), X(LL,MM)
202   CONTINUE
      IF(K.LT.KK) GO TO 130
      GO TO 999
200   WRITE(6,705)
705   FORMAT(1H1,T20,'THE VECTOR IS NOT IN THE STEADY STATE'
     1,' SET, TEST IS GREATER THAN ONE'//T20,
     2'ALF COEFFICIENTS',T45,'TEST VECTOR'//)
      DO 203 LL=1,NN
      WRITE(6,703) ALF(LL), X(LL,MM)
203   CONTINUE
      IF(K.LT.KK) GO TO 130
999   RETURN
      END
```

* *

LIST OF REFERENCES

1. Bartholomew, D. J., Stochastic Models for Social Processes, John Wiley and Sons, Inc., 1967.

2. Branchflower, Norman H., Jr., Analysis of Rank Distributions In a University Faculty, Master's Thesis, Naval Postgraduate School, Monterey, 1970.

3. Thonstad, Tore, Education and Manpower, University of Toronto Press, 1968.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
3.	Commandant (PTP) United States Coast Guard 7th Avenue and D Street, SW Washington, D. C. 20024	1
4.	Assoc. Professor K. T. Marshall, Code 55Mt Department of Operations Analysis Naval Postgraduate School Monterey, California 93940	10
5.	LT Robert L. Armacost USCG 1025 Halsey Drive Monterey, California 93940	5
6.	Professor Robert M. Oliver University of Colorado Boulder, Colorado 80302	1
7.	Department of Operations Analysis, Code 55 Naval Postgraduate School Monterey, California 93940	1

Blank

P. 54

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)* Naval Postgraduate School Monterey, California 93940	2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED 2b. GROUP

3. REPORT TITLE

An Analysis of the Sets of Rank Distributions in An Hierarchical Organization

4. DESCRIPTIVE NOTES *(Type of report and, inclusive dates)*

Master's Thesis; September 1970

5. AUTHOR(S) *(First name, middle initial, last name)*

Robert Leo Armacost

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
September 1970	52	3

8a. CONTRACT OR GRANT NO. b. PROJECT NO. c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) 9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*

10. DISTRIBUTION STATEMENT

This document has been approved for public release and sale; its distribution is unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	Naval Postgraduate School Monterey, California 93940

13. ABSTRACT

A Markov type model for studying rank distribution in an hierarchical organization is examined. Various sets of rank distributions are defined and their properties discussed. A computer aided test is developed for testing a given rank distribution to determine if it is an element of a particular set. For a distribution not in the Steady State set, a test is developed to determine if the distribution can be returned to in m transitions or steps.

DD FORM 1 NOV 65 1473 (PAGE 1)

S/N 0101-807-6811

Security Classification

A-31408

KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Manpower						
Markov Chains						
Personnel						
Rank Distributions						

Thesis
A667
c.1

121593

Armacost
An analysis of the
sets of rank distri-
bution in an hierar-
chical organization.